Take Test: SOA Exam: 11th - 12th May 2022

Test Information

Description This is the exam for the Spring Term 2022 run of CO7214, CO4214, and CO7514 Service-

Question Completion Status:

This exam also serves as the resit exam for those who failed the module in 2021 AND scored less than 50% in the exam.

The exam is available for 24 hours, from Wednesday 11th May, 10 am to Thursday 12th May, 10 am.

This exam does not cover the coursework resit for those who failed the module in 2021 AND scored less than 50% in the coursework. If you are eligible to take the coursework resit, you will have been invited to a separate coursework resit exam during the same 24-hour period.

Instructions This paper contains 8 questions. Full marks can be obtained for answers to all questions.

Please read each question carefully. There are two types of questions:

- "File upload" questions require you to create diagrams in electronic format and upload them.
- "Fill in multiple blanks" should be answered in BlackBoard directly by completing various fields in the question text.

Multiple Attempts This Test allows multiple attempts.

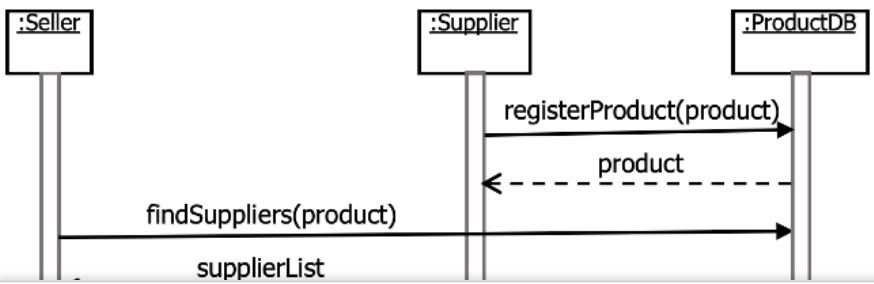
Force Completion This Test can be saved and resumed later.

Your answers are saved automatically.

QUESTION 1

6 points Save Answer

The sequence diagram below shows a scenario where a seller (e.g., an online shop) searches for products in a database where suppliers (e.g., producers or wholesalers) can register products for sale. The task of the team you are leading is to implement the product database.



Click Save and Submit to save and submit. Click Save All Answers to save all answers.

The SOA triangle distinguishes the roles of (1) *requestor*, (2) *provider*, and (3) *registry*.

Task: For each component in this system, indicate which roles it plays in the SOA triangle.

The *Seller* component plays the roles of

Question Completion Status:

The *Supplier* component plays the roles of
and .

The *ProductDB* component plays the roles of
and .

Notes:

- A component can play more than one role. If a component only plays a single role, you can leave the second field empty.
- If you want to assign a single role only to a component, please enter 'none' in the 2nd field.
- If you want to assign two roles to one component, you have to do so in the order in which they are given above, e.g.
  - OK: The ... component plays the roles of *requestor* and *registry.*
  - Not OK: The ... component plays the roles of *registry* and *requestor.*

QUESTION 2 **5 points** Save Answer

Task: Create a use case diagram based on the sequence diagram given under Q1 above.

Attach File

Browse Local Files Browse Content Collection Browse Dropbox

QUESTION 3 **15 points** Save Answer

Task: Create a component diagram at type level, consistent with the sequence diagram given under Q1 above.

Attach File

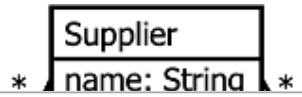
Browse Local Files Browse Content Collection

Click Save and Submit to save and submit. Click Save All Answers to save all answers.

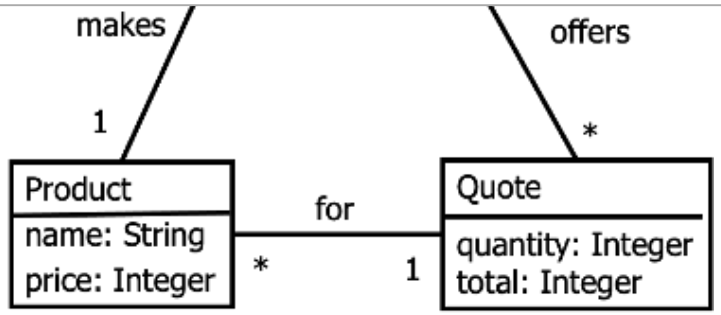
QUESTION 4

16 points Save Answer

The service interfaces use the following conceptual data mo



Question Completion Status:



Task: Detail the interfaces of the components in your component diagram as UML interfaces, including full operation signatures. Where possible use object types as defined in the data model for input and output parameters.

Attach File

Browse Local Files | Browse Content Collection | Browse Dropbox

QUESTION 5

11 points Save Answer

Create a visual contract capturing the following preconditions and effects, consistent with the class and sequence diagrams given above.

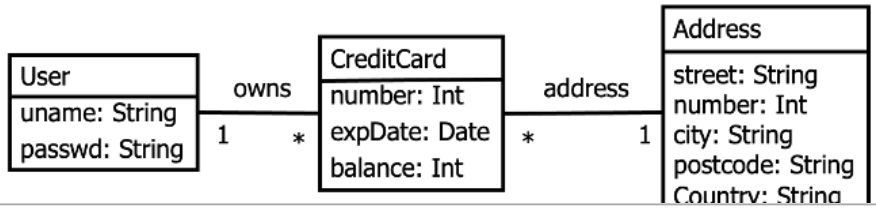
> **Pre:** Supplier *s* received call to *requestQuote(p, qty)* for product *p* by supplier *s*, both present in the supplier's database
>
> **Effect:** Supplier replies with call *provideQuote(q)* where *q* is a new quote for product *p* by supplier *s.*

Use a given function *price(p,qty)* to compute the total price for a quantity *qty* of product *p*.

Attach File

Browse Local Files | Browse Content Collection | Browse Dropbox

Click Save and Submit to save and submit. Click Save All Answers to save all answers.

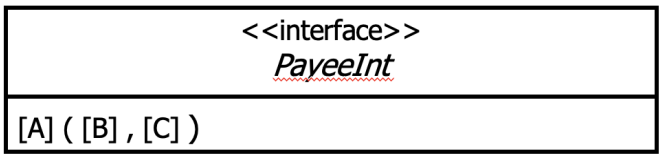


Question Completion Status:

Based on this web service description:

```
<portType name="PayeePortType">
   <operation name="requestPayment">
      <input message="ns:requestPaymentInput"/>
   </operation>
</portType>

<message name="requestPaymentInput">
   <part name="cardNumber" type="cs:int"/>
   <part name="amount" type="xs:int"/>
</message>
<portType name="OwnerPortType">
   <operation name="blockCard">
      <input message="ns:blockCardInput"/>
      <output message="ns:blockCardOutput"/>
   </operation>
   <operation name="getBalance">
      <input message="ns:getBalanceInput"/>
      <output message="ns:getBalanceOutput"/>
   </operation>
</portType>

<message name="blockCardInput">
   <part name="usr" type="xs:string"/>
   <part name="pw" type="xs:string"/>
   <part name="cardNr" type="xs:string"/>
</message>

<message name="blockCardOutput">
   <part name="ok" type="xs:bool"/>
</message>
<message name="getBalanceInput">
   <part name="usr" type="xs:string"/>
   <part name="pw" type="xs:string"/>
   <part name="cardNr" type="xs:string"/>
</message>

<message name="getBalanceOutput">
   <part name="balance" type="xs:int"/>
</message>
```

we want to complete the operation signatures of the following interfaces, such that they are consistent with the XML code and models above.



Click Save and Submit to save and submit. Click Save All Answers to save all answers.

Task:

Complete the *PayeeInt* interface

(, ,)

Question Completion Status:

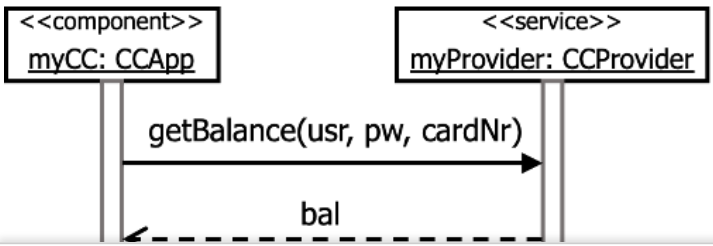
(usr : string , pw : string , cardNr: int): (, , ,):

Notes:

- An operation signature is of the form *operation (parameters) : return* where *parameters* is a comma-separated list of entries of the form *parameter : type* and *return* is an optional return type.
- Make sure to list the operations of each interface in the same order as they are given in the web service description.
- Wherever possible, use the same names for operations, parameters and types as in the web service description, but do not include namespaces such as *xs:* or *cs:*
- Please include a single space between each symbol or word that you enter in each field.

QUESTION 7 **15 points** Save Answer

This sequence diagram describes a sample invocation of one of the operations of the CC service.



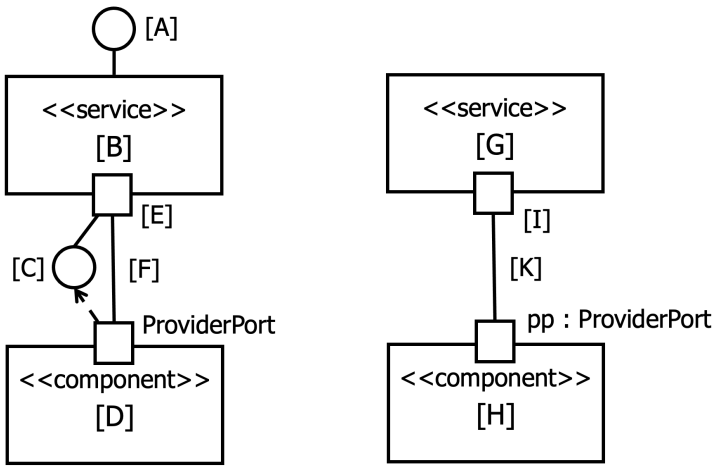
Click Save and Submit to save and submit. Click Save All Answers to save all answers.

```
<service name="CCProviderService">
   <port name="OwnerPort"
         binding="ns:OwnerBinding">
      <soap:address
         location="http://www.ccard.com/owner"/>
   </port>
   <port name="PayeePort"
         binding="ns:PayeeBinding">
     <soap:address
         location="http://www.ccard.com/payee"/>
```

Question Completion Status:

```
<partnerLinkType name=”ownerLinkType">
        <role name=”ownerRole">
                <portType name=”OwnerPortType"/>
        </role>
</partnerLinkType>

<partnerLinks>
   <partnerLink
        name=“ownerLink“
        partnerLinkType="ownerLinkType”
        partnerRole="ownerRole"/>
</partnerLinks>
```

Task: Complete the component diagrams below at type level (left) and instance level (right)



by filling in the blanks corresponding to the letters A-H in the diagrams

- interface names A: _____ and C: _____
- component types B: _____ and D: _____
- port E: _____
- connector type F: _____
- component instances G: _____ and H:

Click Save and Submit to save and submit. Click Save All Answers to save all answers.

- connector K:

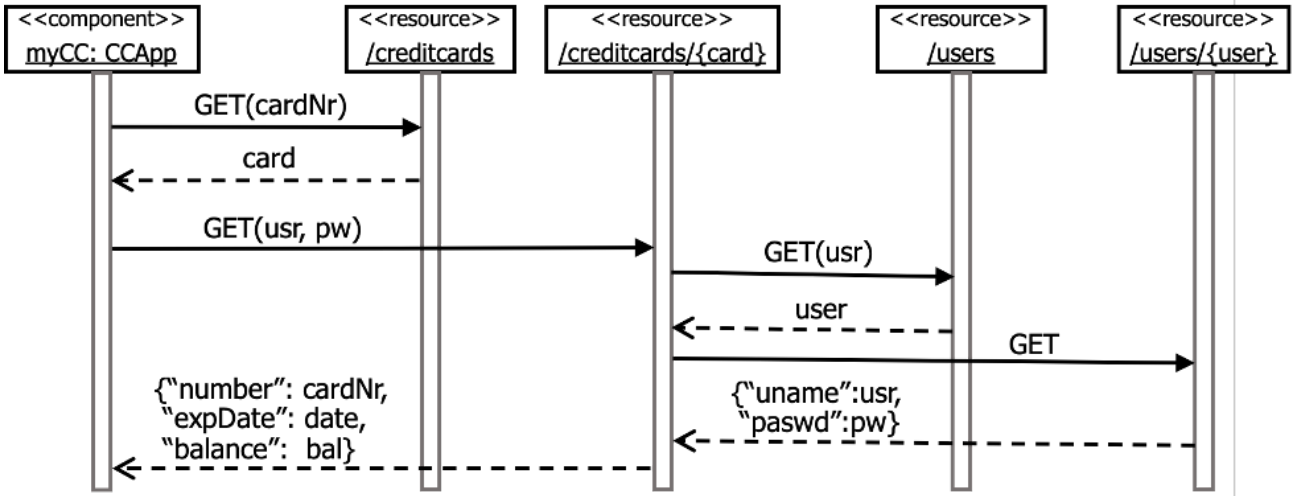
Notes:

- Some of the entries may require several words or symbols.
- Please include a single space between each symbol or word that you enter in each field.

Question Completion Status:

QUESTION 8

20 points Save Answer



Task: Based on this REST-specific sequence diagram implementing the platform-independent sequence diagram of Q7, please complete the OpenAPI specification below by filling in all the blanks.

Notes:

- A field may require more than one word (such as *query* or *string*) or symbol (such as : { } /).
- Please do not include any spaces between words or symbols.
- The matching of answers is case sensitive.

Open API Specification:

```
openapi: 3.0.0
info:
  description: Credit Card Service - noun (CRUD) style
  version: 1.0.0
  title: CCService
paths:
    summary: find card by number
    description: given credit card number, return  id of individual card resource
    parameters:
```

Click Save and Submit to save and submit. Click Save All Answers to save all answers.

```
      required: true
      schema:
        type: string
  responses:
    '200':
      description: OK (200)
      content:
        text/plain:
          schema:
            type: string
```

Question Completion Status:

```
  summary: get card details
  description: if usr and pw match, return the details of the card at the URL
  parameters:
    - name:
      in:
      description: card id
      required: true
      schema:
        type:
    - name:
      in:
      description: user name
      required: true
      schema:
        type:
    - name:
      in: query
      description: password
      required: true
      schema:
        type:
  responses:
    '200':
      description: Card object
      content:
        application/json:
          schema:
            type:
            properties:
              number:
                type: integer
                description: the card number
                type: string
                description: the card expiry date
                type:
                description: the card balance
  summary: find user by user name
```

Click Save and Submit to save and submit. Click Save All Answers to save all answers.

```
    description: user name
    required: true

    schema:
      type:
  responses:
    '200':
      description: OK (200)
      content:
```

Question Completion Status:

```
          type: string
summary: get user details
description: return the details of the user at the URL
parameters:
  - name:
    in:
    description: user id
    required: true
    schema:
      type:
responses:
  '200':
    description: User object
    content:
      application/json:
        schema:
          type: object
          properties:
              uname:
                type: string
                description: the user name
              passwd:
                type: string
                description: the password
```

Click Save and Submit to save and submit. Click Save All Answers to save all answers.